"If towels could tell jokes, they would probably have a dry sense of humor" - Locating and interpreting English puns

Group 2: Sohvi Kalaja, Anssi Moisio, Teemu Pöyhönen, Maristella Spur

November 15, 2020

This paper describes the results obtained by our efforts to find better solutions for the SemEval-2017 Task 7 substasks 2 and 3 than the participating systems in the competition. For homographic puns the best result obtained for subtask 2, pun location, was F_1 score of 0.66 achieved through simple heuristics. None of the additional experiments offered a better result. For subtask 3, pun interpretation, the best result was achieved through the use of word pairs and 4-grams, with the F_1 score of 0.1390, which outperformed the baseline, but not all of the other contestants' systems.

1 Introduction

The aim of this study was to produce a model for the SemEval-2017 Task 7 (Miller et al., 2017) that would successfully detect, locate and interpret puns in English language context. The SemEval-2017 Task 7 consisted of three subtasks. Subtask 1 focused on pun detection, where from a set of English contexts the participants had to detect if a context is a pun or not. Subtask 2 focused on pun location, where from the confirmed pun contexts the systems had to find where in the context is the pun located. Subtask 3 focused on pun interpretation, where the two corresponding senses for the located pun word had to be disambiguated. The goal for this project was to find an improved method for subtasks 2 and 3 by combining features from different competition entries to possibly outperform their results.

In this project we implemented the method developed by Doogan et al. (2017) for subtasks 2 and 3, called the Idiom Savant. The Idiom Savant searched semantic similarities between the two senses of a pun word and grounding words to locate and interpret the pun word. To locate a heterographic pun word, a dictionary of similarly sounding words was planned on using, but due to unforeseen circumstances the part of heterographic pun location had to be left out of the report. By changing a word to a similarly sounding word and searching for frequent idioms from Google n-gram database (Brants

and Franz, 2006) the latent pun word can be identified. In addition to this method, another method created for the competition called Uwaterloo (Vechtomova, 2017) used many scoring methods for locating the pun word. We try to combine this set of criteria with the Idiom Savant to improve the results in the subtask 2. The Python code for this project can be accessed from `https://version.helsinki.fi/anssimoi/lidia-puns`.

2 Related work

According to the Oxford English Dictionary (1989), a pun is described as:

> *The use of a word in such a way as to suggest two or more meanings or different associations, or of two or more words of the same or nearly the same sound with different meanings, so as to produce a humorous effect; a play on words.*
>
> OED Online, Oxford University Press, September 2019

At the same time, though, according to Schröter (2005), *"It is virtually impossible to provide a general definition of pun that gives clear guidelines for distinguishing all puns from all non-puns"* and entries such as the above are not completely free from their own vagueness, considering they might be interpreted differently based on the understanding of the individual reader. More precisely, Miller et al. (2017), who also worked on the SemEval-2017 task, define puns as *"a writer's use of a word in a deliberately ambiguous way, often to draw parallels between two concepts so as to make light of them"*. The 'deliberate ambiguity' implies that such a feature of a word or sentence could be cleared by those people who are exposed to the puns.

Consequently, two factors that seem to be crucial in the processing of a pun are the listeners', or readers', competence in a language and their implicit knowledge of its rules. As explained in Aarons (2017), language is unconsciously reexamined when incongruity and vagueness occur in words or sentences.Thus, she argues that the different *"levels of linguistic representation [...] are largely subconscious, unless we are made aware of them"*. That is to say that, to a certain extent, puns both depend on and, through their comprehension, improve the inner knowledge and competence of an individual in a given language.

Whether consciously or subconsciously, being able to rely on tacit rules and make them explicit is especially relevant in order to obtain valid results in any study concerning pun detection, location, and disambiguation. This can be easily seen in some of the papers written for the SemEval-2017 task, such as JU-CSE-NLP (Pramanick and Das (2017)) and UWaterloo (Vechtomova (2017)), where the authors adopted different self-made sets of criteria that would influence a word's probability of being a pun with the intention of defining guidelines for the wordplay's detection and location in Subtask 1 and 2.

Besides the solutions proposed in the above-mentioned papers, another interesting approach was that of BuzzSaw (Oele and Evang (2017)), which was specifically made to work on the location and interpretation of homographic puns. As mentioned in their report, the interpretation of the pun was done using a knowledge-based WSD (Word

Sense Disambiguation) method, which traditionally relies on the context and the words neighbouring the pun word: the jokes would be divided into two parts, each carrying information regarding the two distinct senses of the pun, and then put into contexts as to determine the most fitting sense in each case.

The SemEval-2017 competition being the source for our own project, studying the reports of the groups that solved the tasks before us was an important preliminary step: by doing this, we could try to understand what could be improved and what methods led to what possible issues and results. Thus, analysing the pros and cons of all the different methods that had already been employed gave us the chance to build our own approach according to what we considered to be the most interesting and relevant solutions for pun location and interpretation.

3 Data

The data sets used in this experiment were provided by the SemEval-2017 competition (Miller et al., 2017). Two different data sets were given, one for homographic and one for heterographic puns. The distribution of the types and subtask division is categorized in Table 1. Homographic puns are puns were the content word has two different meanings for the same spelling of the word, even though the pronunciation might in some cases be different. Examples of homographic puns from the homographic data set:

(1) *I got angry when my cell phone battery died. My counselor suggested I find an outlet.*
(2) *No one chills out in the fires of hell.*

In the case of (1) the pun word would be the word *outlet*, where the two definitions from WordNet (Fellbaum (1998)) would be (n) release, outlet, vent (activity that frees or expresses creative energy or emotion) and (n) wall socket, wall plug, electric outlet, electrical outlet, outlet, electric receptacle (receptacle providing a place in a wiring system where current can be taken to run electrical devices). In the case of (2) the pun is in the word *chill*, where the two meanings are (v) cool, chill, cool down (lose heat) and specifically *chill out* (v) calm, calm down, cool off, chill out, simmer down, settle down, cool it (become quiet or calm, especially after a state of agitation). (Fellbaum (1998)).

Heterographic puns play on the word's pronunciation, where the content word is a word that does not fit into the context normally, but read out loud is pronounced similarly to a word that would fit the context. Examples of heterographic puns from the heterographic data set:

(3) *May the 4th be with you.*
(4) *The nudist defended himself by citing his Constitutional right to bare arms.*

In the case of (3) the reader must have some cultural knowledge to be able to understand the joke. The original line from Star Wars: A New Hope (Lucas (2015)) is "*May the force be with you*", so the pun is realized in the similarity of pronunciation of fourth

[fɔːθ] and force [fɔːs]. The sentence also plays on the two meanings of the word *May*, which can be used in the senses of (n) May (the month following April and preceding June)(Fellbaum, 1998) and in that case referring to the date May 4th, and the auxiliar verb *may* expressing possibility, the intended meaning of the line in the 1977 movie. The context (4) has the pun word *bare*, but in most occasions this context would have the word *bear* due to the Second Amendment in the U.S. Constitution of the right to bear arms. However, both *bare* and *bear* are pronounced the same way [beə].

The data sets were processed differently for each subtask. For subtask 1 there were data set entries both containing and not containing puns, all in a similar format. Most of the non-punning entries were different types of jokes and aphorisms. Each entry containing a pun would only have one pun in the context, no entries with multiple puns were used. All contexts were annotated by different human annotators, and all content words have lexical entries in WordNet 3.1. The pun words were annotated with both two different meanings that it had.

Subtasks 2 and 3 worked with subsets of the same data sets used in subtask 1, where in subtask 2 the contexts not containing a pun were removed, and in subtask 3 the words recognized as pun words were tagged.

pun type	**subtask**	**contexts**	**words**
homographic	detection	2 250	24 499
homographic	location	1 607	18 998
homographic	interpretation	1 298	15 510
heterographic	detection	1 780	19 461
heterographic	location	1 271	15 145
heterographic	interpretation	1 098	13 258

Table 1: Data set statistics (adapted from Miller et al. (2017))

4 Experiments

4.1 Baseline systems and simple heuristics for the pun location subtask

The puns in the data set have largely similar structure which makes it possible to use simple heuristics to guess which word is the pun word. The pun word is

- by definition a content word,
- usually near the end of the context, and
- often an infrequent word.

We implemented a few simple pun locating systems that exploit this common structure of the puns. First, we simply selected the last content word of the context as our guess for the pun word. This could be called the baseline, which gives an F_1 score of 0.60. As a simple heuristic that combines the three facts, we next took the less common word of the last two content words, which gives an F_1 score of 0.66. We used the word frequencies in the Brown Corpus, and the NLTK POS tagger for filtering the content words.

4.2 Idiom Savant for homographic pun location

The first of the more complex methods we implemented is the Idiom Savant for the homographic pun location subtask, developed by Doogan et al. (2017). This system achieved the best results in its task category in the competition. We intended to replicate the system as closely as possible, but the paper does not give all the details so we had to decide some parameters ourselves.

The intuition behind this approach is that the pun word is semantically similar to some of the other words in the context. For example, in the pun "Can honeybee abuse lead to a sting operation?", the pun word *sting* is similar to the grounding words *honeybee* and *operation*. The similarity is measured using Word2Vec embeddings and the cosine similarity function for two vectors:

$$similarity(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{a}||\mathbf{b}|}$$

The Idiom Savant gives each word w_i in the sentence(s) W a score that defines how probable it is that the word is the pun word:

$$score(W, i) = \sum_{j=1}^{n} p_{ij} f_{ij} \sum_{k=1}^{l} \sum_{m=1}^{q} f_{ws} \left(\frac{\boldsymbol{g_k} \cdot \boldsymbol{g_m}}{|\boldsymbol{g_k}||\boldsymbol{g_m}|} \right)$$

where g_j is the jth word in gloss of w_i and g_m is the mth word in gloss of the context word w_j. l and q are the numbers of words in the glosses of w_i and the context word w_j, respectively, and n is the number of words in the context. The glosses are extracted

from WordNet. $p_{ij} = 0.2$ is a constant POS damping factor and $f_{ij} = 0.1$ is a constant frequency damping factor. f_{ws} is a correction factor

$$f_{ws}(x) = \begin{cases} 0, & x < 0.01 \\ 1 - x, & x \geq 0.01 \end{cases}$$

which is supposed to damp the score of when the cosine similarity is too high or too low. This correction gives the largest weight on words with similarity of 0.01 and decreases linearly as x increases, depicted in Figure 1. Doogan et al. (2017) do not explain the

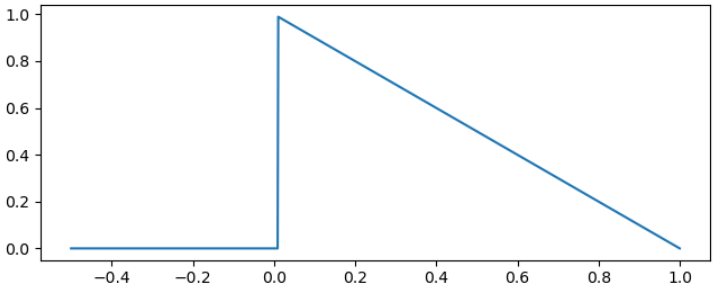

Figure 1: $f_{ws}(x)$

choice of this function. Frankly, this correction function does not make much sense because it undermines, or reverses, the similarity score. For example, if the similarity score were 0.7, the "correction" function would yield 0.3, and if the similarity score were 0.3 it would yield 0.7. A possible explanation to this might be a typing error in the paper, because it seems somewhat unlikely that they used this function. If the idea is to dampen the similarity scores that are too high or too low, a better function might be something like the following:

$$f_{ws}(x) = \begin{cases} 0, & x < 0.01 \\ 0, & x \geq 0.90 \\ x, & otherwise \end{cases}$$

Furthermore, the scoring function does not normalise the sums of the similarities in any way. This means that if a word happens to have a long definition in WordNet, a larger number of cosine similarity values is summed together. We added normalisation factors $\frac{1}{lq}$ in our implementation to control for the differences in the lengths of the word definitions (i.e., glosses in WordNet):

$$score(W, i) = \sum_{j=1}^{n} p_{ij} f_{ij} \frac{1}{lq} \sum_{k=1}^{l} \sum_{m=1}^{q} f_{ws} \left(\frac{\boldsymbol{g_k} \cdot \boldsymbol{g_m}}{|\boldsymbol{g_k}||\boldsymbol{g_m}|} \right)$$

We were not able to get good results using this method. All the results were worse than our baseline of $F_1 = 0.60$. Furthermore, the method is computationally expensive,

so we did not calculate results for the full data set but only a subset with about 10% of the data set. We were not able to achieve results that are even close to those reported by Doogan et al. (2017). This can be because of any of the details that they did not write about in the paper and we could not therefore replicate. Most significant detail is probably the fall-back system. If the scoring function does not give a clear suggestion for the pun word, the system should fall back to some baseline, e.g. the last word. Alternatively, the last word(s) should be given some additional weight in the scoring function.

4.3 Homographic pun interpretation

For the pun interpretation task, we came up with our own approach based on the WSD system by Vial et al. (2019). The system disambiguates words using a pretrained BERT (Devlin et al., 2018) transformer and has recently achieved state-of-the-art results in many WSD evaluation competitions[1]. Our approach is to disambiguate the pun multiple times using different subsets of the context words and select the two senses that appear most often. First we paired each word of the context with the pun word, for example the pun "If you're a gardener, you might call yourself a plant manager":

```
if plant
you plant
re plant
a plant
gardener plant
you plant
might plant
call plant
yourself plant
a plant
plant manager
```

When each line is disambiguated separately, we get different senses for the pun word "plant":

```
if plant|plant%1:03:00::
you plant|plant%2:35:00::
re plant|plant%1:06:01::
a plant|plant%1:03:00::
gardener|gardener%1:18:00:: plant|plant%1:03:00::
you plant|plant%2:35:00::
might|might%1:07:00:: plant|plant%1:03:00::
call|call%2:32:02:: plant|plant%1:06:01::
yourself plant|plant%1:06:01::
```

[1]For example `https://paperswithcode.com/paper/sense-vocabulary-compression-through-the#code`

```
a plant|plant%1:03:00::
plant|plant%1:06:01:: manager|manager%1:18:00::
```

The idea is that the pun word would get the two desired (correct, i.e., plant%1:06:01:: and plant%1:03:00::) senses when it is paired up with its two (or more) grounding words, in this example the words "gardener" and "manager". With this method we get an F_1 score of 0.1061. The result would have been in 4th place in the competition. However, it is still worse than the most frequent sense baseline result reported by Miller et al. (2017), which is 0.1348.

Next, we tried adding also n-grams that include the pun word for the WSD system to disambiguate. Using trigrams, we would add the lines

```
yourself a plant
a plant manager
```

to the example pun. This improves the results as shown in Table 2.

	MFS baseline	Duluth (DM)	Miller & Gurevych	Word pairs	Word pairs + trigrams	Word pairs + 4-grams	Word pairs + 5-grams
coverage	1.0000	0.8606	0.6826	0.7427	0.8629	0.8729	0.8806
precision	0.1348	0.1683	0.1975	0.1244	0.1313	0.1491	0.1452
recall	0.1348	0.1448	0.1348	0.0924	0.1133	0.1302	0.1279
F_1	0.1348	0.1557	0.1603	0.1061	0.1216	0.1390	0.1360

Table 2: Results of the MFS baseline and the best systems reported by Miller et al. (2017), and our results.

We get the best results using the word pairs and 4-grams, which is better than the most frequent sense baseline, but not as good as the best results in (Miller et al., 2017).

5 Conclusion

We implemented some systems for the homographic pun location task and the homographic pun interpretation task. Our task is not comparable to the SemEval competition, because we used the test data as evaluation data. This makes the task easier because we could tune our system using the test data.

In the homographic pun location task, we got the best results using a simple heuristic of selecting the less common of the two last words. Vechtomova (2017) used similar heuristics in their system. With this method, we got an F_1 score of 0.66, which is better than all of the participant systems except the Idiom Savant, which got F_1 score of 0.66 (about 0.005 better than our result). We implemented the Idiom Savant for homographic puns, but did not get similar results as Doogan et al. (2017).

We developed an approach for the homographic pun interpretation task. Our idea was that disambiguating multiple different subsets of the pun text would yield the two senses for the pun word because the subsets include different grounding words that prompt the

two different senses for the pun word. Some of the competition participant systems used similar methods. For example, Oele and Evang (2017) divided the pun text into different contexts and disambiguated them separately, similarly to our method.

The interpretation subtask is difficult, and none of the systems in the competition exceeded the baseline result by a large margin. Similarly, we got just a little better result than the most frequent sense baseline, shown in Table 2.

References

Aarons, D.
2017. Puns and tacit linguistic knowledge. In *The Routledge handbook of language and humor*, Pp. 80–94. Routledge.

Brants, T. and A. Franz
2006. Web 1t 5-gram version 1 (2006). *Linguistic Data Consortium, Philadelphia.*

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova
2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805.*

Dictionary, O. E.
1989. Oxford english dictionary. *Simpson, JA & Weiner, ESC.*

Doogan, S., A. Ghosh, H. Chen, and T. Veale
2017. Idiom savant at semeval-2017 task 7: Detection and interpretation of english puns. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, Pp. 103–108.

Fellbaum, C.
1998. Wordnet: An electronic lexical database. 1998. *WordNet is available from http://www. cogsci. princeton. edu/wn.*

Lucas, G.
2015. *Star Wars, Episode IV: A New Hope.* Fox-Paramount Home Entertainment (Denmark).

Miller, T., C. Hempelmann, and I. Gurevych
2017. Semeval-2017 task 7: Detection and interpretation of english puns. Pp. 58–68.

Oele, D. and K. Evang
2017. Buzzsaw at semeval-2017 task 7: Global vs. local context for interpreting and locating homographic english puns with sense embeddings. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, Pp. 444–448.

Pramanick, A. and D. Das
2017. Ju cse nlp@ semeval 2017 task 7: Employing rules to detect and interpret english puns. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, Pp. 432–435.

Schröter, T.
2005. *Shun the Pun, Rescue the Rhyme?: The Dubbing and Subtitling of Language Play in Film.* PhD thesis, Estetisk-filosofiska fakulteten.

Vechtomova, O.
2017. Uwaterloo at semeval-2017 task 7: Locating the pun using syntactic characteristics and corpus-based metrics. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, Pp. 421–425.

Vial, L., B. Lecouteux, and D. Schwab
2019. Sense vocabulary compression through the semantic knowledge of wordnet for neural word sense disambiguation.